Supervised learning for Human Action Recognition from multiple Kinects

Wang Hao, East China Normal University (ECNU), Chine
Christel Dartigues, Université Côte d'Azur, UMR CNRS 7271 I3S, Nice
Michel.Riveill, Université Côte d'Azur, UMR CNRS 7271 I3S & Equipe Inria MAASAI, Nice

1 Abstract

The research of Human Action Recognition (HAR) has made a lot of progress in recent years, and the research based on RGB images is the most extensive. However, there are two main shortcomings: the recognition accuracy is insufficient, and the time consumption of the algorithm is too large. In order to improve these issues our project attempts to optimize the algorithm based on the random forest algorithm by extracting the features of the human body 3D, trying to obtain more accurate human behavior recognition results, and can calculate the prediction results at a lower time cost. In this study, we used the 3D spatial coordinate data of multiple Kinect sensors to overcome these problems and make full use of each data feature. Then, we use the data obtained from multiple Kinects to get more accurate recognition results through post processing.

2 Introduction

Human Action Recognition (HAR) is an active research topic in Computer Vision and a very popular and useful task in various fields. It especially plays an important role in people's daily life. Human fall detection systems are very often needed for many people in today's aging population including the elderly and people with special needs such as the disabled, as fall is the main cause of injury-related death for elderly people [1, 2]. Automatic detection of human fall is then a key issue in health management systems. At the same time, HAR is also used in smart home, security video surveillance security and Tele-immersion System. Different approaches are used to build human fall detection systems, including wearable based devices, non-wearable sensors, and vision-based system. Wearable based devices such as accelerometers and gyroscopes are highly preferred by engineers and doctors [3, 4, 5]. However, methods based on those equipment have some shortcomings due to the lack of understanding of context and the ability to extract information features [6]. Wearable devices often generate too many false alarms, and wearable devices can also cause inconvenience to people's lives, resulting in a reduced willingness for elderly equipment. There are also sensor devices that do not need to be worn, installed in a room environment such as floor vibration sensors. These non-wearable devices eliminate the trouble of wearing, but it is still difficult to satisfy people on accuracy. Therefore, the scheme based on visual devices such as cameras has become an applicable choice

because it can acquire more human motion information and has a wide range of detection. For these reasons, vision-based devices have higher accuracy in the daily behavior classification of the human body. In the past, there were many works based on vision-based devices for human action recognition. But due to the influence of variations of people, illumination and viewpoint, activity phase and occlusions, there will still be more false positives [7, 8, 9, 10].

The emergence of Microsoft Kinect has opened up new opportunities for solving these problems. The Kinect sensor combines a special infrared light source to capture depth information. Meanwhile, Kinect's SDK can generate human skeleton data. RGB data can provide important features of the human body's appearance, but also has a larger range of acquisition. However, the calculation of RGB data features always requires a lot of time, which is not well adapted to the needs of daily life. To avoid this problem, we mainly use Skeleton data and depth data, as it helps to more accurately identify human actions. Skeleton data is mainly composed of scalar vectors, and the calculation speed is thus very fast. Kinect sensors are also limited by the measurement angle and distance range, and they are also affected by noise: people may exceed the monitoring range when falling, resulting in unsatisfactory action recognition. In order to solve this problem, many studies consider multiple multiple Kinect to capture human action from different angles and distances. We can then integrate the data obtained for the final prediction results.The prediction is done thanks to a learning algorithm. In our research, we developed a successful approach based on Random Forest [11].

We will first discuss about existing HAR works and we will present the learning algorithm on which we based our research (Random Forest). Secondly, we will present our methodology and we will present in a third part the dataset we choose and our experiments.

3 Related Work

3.1 RGB-Based work

The academic community has rich research on human action recognition. We mentioned solutions mainly for wearable devices, non-wearable environmental sensors, and vision-based devices. We mainly discuss human body recognition based on visual information of RGB cameras or Kinect cameras to obtain accurate contour and depth information of the human body. Among all the existing methods, extracting features using RGB image data is the most popular approach. The RGB camera is inexpensive, and it has also spawned many datasets based on RGB images. Most of the methods based on RGB image detection of human action first need to detect the human body area, draw a border of the human body contour and then extract the behavior characteristics of the human body in the border. The work in [7] proposed to use variations in silhouette area that are obtained from only one camera. They use a simple background separation method to acquire the silhouette and find that the proposed feature is view-invariant. And the work in [8] used Support Vector Machine (SVM)

for classification. The foreground human silhouette is extracted via background modeling and tracked throughout the video sequence. The human body is represented with ellipse fitting. Then, the shape deformation quantified from the fitted silhouettes is used as the features to distinguish different postures of the human. Finally, they classify different postures via a multi-class SVM and a context-free grammar-based method. The work in [12] tried to estimate 3D human pose from a sequence of monocular images. This paper presents a Recurrent 3D Pose Sequence Machine(RPSM) to automatically learn the image-dependent structural constraint and sequence-dependent temporal context by using a multi-stage sequential refinement. And get better results on Human3.6m [13] and HumanEva-I dataset [14].

Joao Carreira, Andrew Zisserman used deep Convolutional Networks (ConvNets) in 2014 to identify human action in the video [9]. They attempted to capture the complementary information on appearance from static frames and motion between frames. A dual-stream ConvNet architecture with spatial and temporal networks was proposed. Under the wired training data, ConvNet trained on multi-frame dense optical flow can achieve excellent performance: 88% accuracy was obtained on UCF-101 dataset [10] and 59.4% accuracy on HMDB-51 dataset [15]. They did further work [16] based on this. The original model was upgraded, and the 3D convolutional neural network can be constructed by computing features from both spatial and temporal dimensions. The training was re-trained on the new training set Kinetics Human Action Video dataset. The result achieved 97.9% accuracy on UCF-101 dataset and 80.2% on the HMDB-51 dataset. The use of convolutional neural networks requires high hardware(GPU) and extended training time. It needs to adjust parameters to get the best model. The prediction accuracy may not be guaranteed after replacing new dataset.

3.2 Skeleton-based work

With the advent of the Kinect camera, the efficient RGB-D sensor provides a new direction for human action recognition. In addition to RGB graphics, the Kinect camera provides depth and skeleton information independent of lighting conditions. In [17], the author used the depth pattern to extract human body image boundaries. Then they calculated the curvature dimension spatial characteristics of the human contour and applied the extreme learning machine to classify the different actions. The work in [18] used the hierarchical recurrent neural network (RNN) to perform motion recognition on 3D skeleton data. They divided the human skeleton into five parts according to the human physical structure and then separately feed them to five RNN subnets. They get excellent performance, but this method encounters overfitting problems. In [19], the authors chose a set of key-pose-motifs for each action class. They classified a sequence by matching it to the motifs of each class and selecting the class that maximizes the matching score. The work in [20] used an angular representation of the skeleton joints to describe each pose. They used those descriptors to identify key poses through a multi-class SVM. The gesture is then labeled from the key pose sequence with a decision forest.

For the fall detection problem, there are more specific options to choose. Two fall detection algorithms are proposed in [21] . One determines whether a drop has occurred by a single frame. The second uses time-series data to distinguish falls and slowly lay down on the floor. In [22] , they tried to explore secondary features (angle and distance), focusing on the correlation between joints and the boundary of this correlation. The authors mainly focused on the angle of the joints on the legs, and the distance from the floor to several important joint points. The algorithm is simple, but the prediction results are unstable due to the quality of joint tracking. Author of [23] considered that Kinect could not track all joints correctly. They defined and computed three features (distance, angle, velocity) on only several vital joints. Then they used SVM to analyzed ten specific actions with good results.

Trying to combine RGB data with skeleton data is also an effective method. There is a novel method in [24] which uses skeleton data to obtain the 3D bounding box of the human body. It then measures the velocity based on the contraction or expansion of the width, height, and depth of the 3D bounding box. The authors of [25] creatively installed the camera on the ceiling. The human head-to-ceiling distance is an important feature that combines the application of the accelerometer with the K-Nearest-Neighbour (KNN) classifier for identification. The work in [26] used a tri-axial accelerometer to indicate the potential fall as well as to indicate whether the person is in motion. If the measured acceleration is higher than an assumed threshold value, the algorithm extracts the skeleton, calculates the features and then executes the SVM-based Classifier to authenticate the fall alarm. A similar work in [27] is also using KNN, where an accelerometer is used to indicate a potential fall, and the Kinect sensor is used to authenticate an eventual fall alert. Only the depth image captured during the possible fall is processed.

It is also quite skillful to know how to combine the information obtained by multiple cameras. Earlier fusion and late fusion are mentioned. The work in [28] tried to use a new cross-view action representation. They propose a method effectively express the geometry, appearance, and motion variations across multiple viewpoints with a hierarchical compositional model. They used 3D skeleton data acquired from Kinect to train, Northwestern-UCLA Multiview Action3D Dataset and dataset MSR-DailyActivity3D Dataset [29]. Then they tested the model with unknown 2D video. They succeeded to use the cross-view to improve the accuracy and robustness of action recognition.

We mainly focus on the work in [30] using the combination of RGB and skeleton data. They installed seven Kinects with different angles in the room. They used the skeleton data to obtain the vertical velocity and the height of the human body from the ground. If the Kinect tracking fails and does not generate enough skeleton data, the features are extracted from the continuous RGB data. The human action recognition is performed based on the SVM classifier. Finally, the results of the seven Kinect data processing are combined. This method also achieved an accuracy of 91.5%, but we can observe two main problems: processing RGB images still require a lot of calculations and skeleton data are not

used enough. In many cases, the use of RGB images tends to cause users' concerns about privacy issues. All those studies show that using skeleton instead of purely RGB data is a good solution. Combining the skeleton with another classifier such as the Random Forest also shows interesting results [11] on the MSR-DailyActivity3D Dataset. In this study the vector representing a moment in the flow of the data is composed of all the coordinate of the joints of the skeleton and all the distances and angles between the joints. Several consecutive moments are combined in one vector in order to fully describe an action. Thanks to all those works and especially the last one, we choose in this study to consider Random Forest as main classifier for our work.

4 Technical overview

In this section, we describe the basic concepts and characteristics of Microsoft Kinect camera and explain the skeleton data generated by this equipment. We then explain the principle of the Random Forests and explai why they are suitable for our human behavior recognition task.

4.1 Microsoft Kinect

Kinect is a motion-sensing input device by Microsoft for the Xbox360 video game console and Windows PCs. Kinect with full skeleton mode can track a person's actions and generate 20 joints as the skeleton data [31]. Each joint include the value of (x,y,z) in 3-dimensional space. Figure 1 shows the 20 joints of the human body. Our work uses the data generated by full skeleton mode.

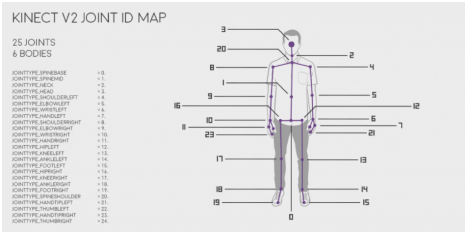

Fig. 1. 20 joints generated by Kinect

The skeleton data obtained from Kinect is 20 key joints of the body. Each joint is a 3- dimensional vector. The data volume of 20 joints is not enough to support the requirements of training data for machine learning algorithms. We refer to the method in [26] to calculate angles, distances, and other information through different joints. In this way, more human body motion features can be extracted, and the amount of data is greatly enriched, which is very helpful for the classification algorithm.

4.2 Random Forest

Random Forest have been first formally introduced in [32]. In this paper, Leo Breiman define a Random Forest as a multi-classifier composed of a set a decision trees.The method defined by Breiman is called Forest-RI (Random Forest - Random Input) and is still a very popular approach.

In our work, we use the R-package Random Forest v4.6-14 to implement the Random Forest. This package implements Breiman's Random Forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression.

5 Methodology

In this section, we present our approach of classification of HAR from a multi-Kinect dataset with Random Forest. We first describe the vector created from the raw dataset. We then describe our innovative approach based on two important points: the cutting of the skeleton into five significant subparts and the development of a hierarchical Random Forest algorithm. We will end this part by describing how we managed the multi-views data.

5.1 Feature vector

Since we only use 3D skeleton data, we first consider 20 joints (characterized by (x, y, z) coordinates) in a three-dimensional vector. This vector doesn't contain enough data to fully classify different actions. Similarly to the work done in [11], we have augmented our feature vector by compute and add all possible distances/angles between all possible pair/triplet of joints. This process ended to a feature vector of 3610 values: 3420 angle values followed by 190 distance values.

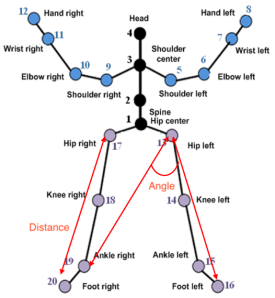

Fig. 2. Distance and Angle of skeleton joints

As shown in Figure 2, a distance data can be generated between any two joints, and three angle vectors can be calculated between any three joints. This

feature vector will be represented as FV in this paper. Each frame can generate one-row FV like Figure 3.

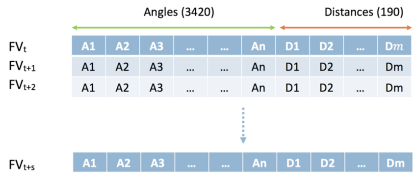

Fig. 3. Feature Vector of Angles and Distances

At the same time, the action of the human body is a dynamic process; we need to acquire the temporal features of an action at the same time. For each frame with time index t, we extract the pairwise relative position features by taking the difference between the position of joint at time t and that of others frame with time index t+1:

$$Diff2_t = FV_{t+1} - FV_t \tag{1}$$

This new feature $Diff2_t$ provides some temporal information by varying the angle and distance between two consecutive frames. But calculating the difference between two adjacent frames is not robust. We need to rich temporal information. In the work of [33, 34], there is a Spatial Pyramid approach. We set a 10-frame sliding window, and calculate the difference between different number of frames, 2,5 and 10 frames. Then we get the $Diff5_t$ $Diff5M_t$ $Diff10_t$ represent the action movements in 5 and 10 frames.

$$Diff5_t = FV_{t+4} - \sum_{i=t}^{i=t+3} FV_i \tag{2}$$

$$Diff5M_t = FV_{t+8} - \sum_{i=t+4}^{i=t+7} FV_i \tag{3}$$

$$Diff10_t = FV_{t+9} - \sum_{i=t}^{i=t+8} FV_i \tag{4}$$

In a 10-frame sliding window, the differences between each pair of consecutive poses will be sum up into two Diff5 feature vectors, one from t to t+4 and another one from t+5 to t+9. In order to preserve the coverage we had with the Diff2 feature vector and the overlapping sliding window. We also define a Middle Diff5 feature vector by calculating the middle 5 position and averaging the Diff2 at these positions [11]. The Figure 4 shows the consistency of each Diff feature vector.

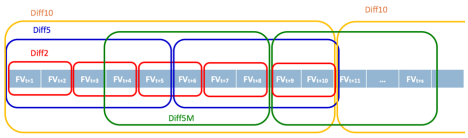

Fig. 4. Sliding Window and Definition of Diff2,Diff5,Diff5M,Diff10 Feature Vector

Using 10 frames we can obtain 9 Diff2, 2Diff5, 2Diff5M and 1 Diff10 features. At last, we can get 46930 values for every 10 frames. We will use this FV-46930 as the input of the Random Forest. The acquisition rate of CMDFALL is 20 Hz. Therefore, the Kinect camera will capture 20 frames in one second. We set 10 frames as the sliding window, which is reasonable because most fall action in the dataset are completed in 0.5 seconds.

5.2 Decompose whole body data into subparts

Training a Random Forest with whole-body data points does not necessarily yield good results because some human actions do not lead to whole-body movements. Moreover, this leads to very long computation time. In order to reduce preprocessing time and learning time we decompose the human body into five distinct subparts: left arm, right arm, left leg, right leg and the upper of the body. On our server machine (2 Processors Intel Xeon X5675 at 3,06GHz and 24GB RAM), calculating the feature vectors (distances and angles) by whole body (20 joints) from more than 400 files will take more than 24 hours. In subpart mode, it only consumes few minutes.

For each subpart of the body, a Random Forest is build. Each subpart contains 4 joints, and we also calculate the distances and angles. The 4 joints could generate a feature vector with 6 distance values and 12 angle values. For each subpart-Random Forest, we calculate a prediction score and normalize the five obtained scores to get a percentage of the prediction. Further, we connect the prediction percentage generated by each subpart-Random Forest with the distances/angles vector to form a new feature vector (FV-190) with 190 values. Then we use the FV-190 to build a new Random Forest. The new feature vector is shown in Figure 5.

Figure 6 shows the entire process of Hierarchical Random Forests. At first, we decompose the human body into five subparts. Secondly we use partial skeleton data to build five subpart-Random Forests and obtain five prediction scores. We then compose the new feature vector by concatenating the prediction scores obtained in the preceding step and the distances/angles feature of each subpart. A new Random Forest is finally built thanks to this new vector to obtain the final decision result.

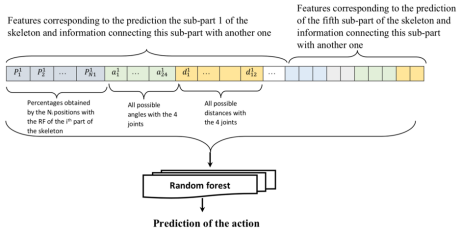

Fig. 5. New feature vector with 190 values

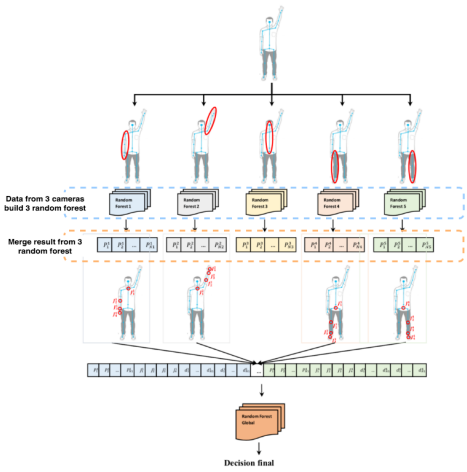

Fig. 6. Process of hierarchical random forest

5.3 Multi-views skeleton data

For this study we restricted ourselves to 3 Kinect cameras installed from 3 different angles: skeleton data of the same action are recorded by 3 Kinect cameras at the same time. We can easily generalized to more than 3 Kinects. Integrating the 3 views skeleton data can help us to improve the precision of action recognition. We provide three strategies and validated the predictions of these different strategies through experiments.

Strategy 1: Merge 3 Kinect vectors into 1 vector and build 1 RF

In our first strategy, we consider the feature vectors generated by each Kinect and we merge those 3 vectors into one. We then get a bigger feature vector with 140790 values. In this way each input sample is characterized by more features. We use all the obtained vectors for our dataset and use them to build a single Random Forest.

Strategy 2: Merge 3 Kinect vectors in 3 vectors and build 1 RF In this strategy, also called early fusion strategy, we start with the three feature vectors generated each Kinect and we use all of them to build only one dataset. This

dataset is then used to build a single Random Forest. In this way, multi-views only able to add the training samples.

Strategy 3: Consider 3 Kinect vectors and build 3 RFs The last strategy, also called late fusion strategy, consists in building a dataset for each Kinect specific vector. Each dataset is then used to build a Random Forest and the decisions of each RF are then combined.

6 Experiment

We test different steps of our algorithm on the CMDFALL dataset and we compared our result to state-of-art methods in [30].

6.1 Dataset & Setup

We use CMDFALL as the data set for our experimentation to test our approach. Calculating the distance and angle of the skeleton and calculating the difference between consecutive frames requires a lot of calculations. We thus perform operations on a server. The hardware configuration is: Dell PowerEdge R710 Rack, 2 Processors Intel Xeon X5675 at 3,06GHz, 6 Cores, 12MB cache memory, 24GB RAM DDR3-1333MHz, 2 hard disks Hot Plug 600GB SAS 6Gbit/s 15000tr/min with RAID 0 for performance. At our data scale, building the Random Forest does not need high hardware requirements. Building a Random Forest on the subparts of the skeleton takes about a few minutes, and it takes about one to two hours to build a Random Forest of the whole body skeletons.

6.2 Whole body skeleton

We test the whole-body mode at first. We use the 20 joints skeleton data to build the model and fuse the data from the three cameras in the different ways corresponding to our three strategies.

Strategy 1: Merge 3 Kinect vectors in 3 vectors and build 1 RF In the first method, we merged the data acquired by the three Kinects into a single vector. Each Kinect's data for each frame generates a vector of 46,930 values that are combined into a vector of 140,790 values. 5 and 10 are used as parameters of the sliding window, respectively. Since the calculation time is too long, we do not calculate the number of sliding windows smaller than 5. At the same time, we used different number of trees for the forest: 500, 1000 and 1500. The specific results are shown in Table 1.

tree number	500	1000	1500	2000
Sliding window 10	37.43%	36.05%	33.84%	39.49%
Sliding window 5	29.43%	28.14%	27.98%	27.30%

Table 1. Class error rate of 3 Views in one row

Result in Table 2 shows there are not obvious improvement for class classification accuracy. But precision of detection the fall action improves a lot. Only for fall action class, the precision is about 90

tree number	500	1000	1500	2000
Sliding window 10	35.23%	32.65%	31.06%	35.88%
Sliding window 5	29.58%	27.65%	27.78%	25.58%

Table 2. Merged Class error rate of 3 Views in one row

Strategy 2: Merge 3 Kinect vectors in 3 vectors and build 1 RF In the second method, we treat all the views side by side. Three batch of skeleton data from different Kinects will be used as input samples together. Similarly, we get the following results in Table 3. There is no obvious improvement in the results, even worse. According to the analysis of the classification results, the same actions from different views are not easily classified into same class. Explain that the feature values of the actions captured by different views obtained by our method have large differences. We believe that there should be better means to obtain more relevant feature from different views.

tree number	500	1000	1500	2000
Sliding window 10	40.52%	39.10%	39.83%	38.78%
Sliding window 5	38.26%	36.99%	37.54%	25.58%

Table 3. Class error rate of 3 Views in 3 rows

Strategy 3: Consider 3 Kinect vectors and build 3 RFs In the third method, we build three independent Random Forests using data from different Kinects. After the test data input, we add the predicted scores of the three Random Forest outputs, and the highest score is the final result. The final classification results are shown in the Table 4. The superposition of their respective prediction results from 3 independent Random Forests, did not significantly improve the precision rate. According to the detailed classification results, in some cases, the 3 Random Forests will be wrong in predicting the same class.

tree number	500	1000	1500	2000
Sliding window 10	34.13%	32.28%	37.64%	36.43%
Sliding window 5	28.23%	27.64%	26.12%	28.04%

Table 4. Class error rate of 3 Views defining 3 Random Forests

6.3 Subpart body & Hierarchical Random Forest

Using whole body skeleton data does not get ideal result. In order to achieve higher accuracy, we try to break down the human body into 5 parts, to build the five sub-Random Forests separately and to integrate the prediction results and combine them to build a new Random Forest (Hierarchical Random Forest). We tested each with a sliding window of 10. Number of trees in random forest is 500. We randomly choose X% samples as training set and (1-X) % as testing set. Using training set to define the Hierarchical Random Forest and input the testing set to the global Random Forest to get the prediction result. The results are much better than for the whole body. The result is shown in Table 5. We set X as 80, 60 and 40. Kinect3, Kinect4 and Kinect5 means the data is captured by the Kinect cameras 3, 4 and 5.

Training Percent	80%	60%	40%
Kinect3	99.10%	97.08%	95.40%
Kinect4	99.55%	97.60%	97.40%
Kinect5	99.10%	96.81%	95.82%

Table 5. Classification precision of Hierarchical random forest

When using 80% dataset as training set, we could get precision over 99% in individual view. And our method is simpler and faster in computing, using only skeleton data. It consumes about 5 minutes to calculating feature vector (distances and angles) from 418 files, each files. Building random forest will take less than 5 minutes.

7 Conclusion

This paper proposes a new human action detection method that uses only three-dimensional skeleton data. Without the using RGB images or motion velocity information collected by other wearable sensors such as accelerometer and gyroscopes. Using only skeleton data can reduce the amount of computation, while also avoiding the troubles of wearing devices. We constructed a Hierarchical Random Forest with five subparts of the whole body skeleton decomposition. Subpart mode effectively reduce the time consumption of traditional algorithms and Hierarchical Random Forest greatly improve the precision of classification. We get the most static features by calculating all possible angles and distances between each joint. The temporal characteristics are extracted by calculating the difference directly between adjacent frames. We tested our approach on the CMDFALL data set and the results are satisfactory for the whole body but they are better with the subparts combined with the Hierarchical Random Forest, with an average classification precision 98.5% on CMDFALL.

Acknowledgements. This work was supported by NSFC grants(No.61532021 and 61972155).

References

[1] Clare Griffiths, Cleo Rooney, and Anita Brock. Leading causes of death in england and wales–how should we group causes. *Health statistics quarterly*, 28(9), 2005.

[2] Yoosuf Nizam, Mohd Norzali Haji Mohd, and M Mahadi Abdul Jamil. Classification of human fall from activities of daily life using joint measurements. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8(4):145–149, 2016.

[3] AK Bourke, JV Obrien, and GM Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, 26(2):194–199, 2007.

[4] Fabio Bagalà, Clemens Becker, Angelo Cappello, Lorenzo Chiari, Kamiar Aminian, Jeffrey M Hausdorff, Wiebren Zijlstra, and Jochen Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one*, 7(5):e37062, 2012.

[5] Che-Chang Yang and Yeh-Liang Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10(8):7772–7788, 2010.

[6] Luis Gioanni, Christel Dartigues-Pallez, Stéphane Lavirotte, and Jean-Yves Tigli. Using random forest for opportunistic human activity recognition: a complete study on opportunity dataset. In *11èmes journées francophones Mobilité et Ubiquité, Ubimob 2016, Lorient, France, July 5, 2016.*, 2016.

[7] Behzad Mirmahboub, Shadrokh Samavi, Nader Karimi, and Shahram Shirani. Automatic monocular system for human fall detection based on variations in silhouette area. *IEEE Trans. Biomed. Engineering*, 60(2):427–436, 2013.

[8] Weiguo Feng, Rui Liu, and Ming Zhu. Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. *Signal, Image and Video Processing*, 8(6):1129–1138, 2014.

[9] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 568–576, 2014.

[10] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[11] A. Aly Halim, Christel Dartigues-Pallez, Frédéric Precioso, Michel Riveill, Abderrahim Benslimane, and Salma A. Ghoneim. Human action recognition based on 3d skeleton part-based pose estimation and temporal multi-resolution analysis. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*, pages 3041–3045, 2016.

[12] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3d pose sequence machines. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5543–5552, 2017.

[13] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, July 2014.

[14] Leonid Sigal and Michael J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, 2006.

[15] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2556–2563, 2011.

[16] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733, 2017.

[17] Xin Ma, Haibo Wang, Bingxia Xue, Mingang Zhou, Bing Ji, and Yibin Li. Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE J. Biomedical and Health Informatics*, 18(6):1915–1922, 2014.

[18] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1110–1118, 2015.

[19] Chun-yu Wang, Yizhou Wang, and Alan L. Yuille. Mining 3d key-pose-motifs for action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2639–2647, 2016.

[20] Leandro Miranda, Thales Vieira, Dimas Martínez Morera, Thomas Lewiner, Antônio Wilson Vieira, and Mario Fernando Montenegro Campos. Online gesture recognition from pose kernel learning and decision forests. *Pattern Recognition Letters*, 39:65–73, 2014.

[21] Christopher Kawatsu, Jiaxing Li, and C. J. Chung. Development of a fall detection system with microsoft kinect. In *Robot Intelligence Technology and Applications 2012 - An Edition of the Presented Papers from the 1st International Conference on Robot Intelligence Technology and Applications, RiTA 2012, Gwangju, Korea, December 16-18, 2012*, pages 623–630, 2012.

[22] Martha Magali Flores-Barranco, Mario Alberto Ibarra-Manzano, and Irene Cheng. Accidental fall detection based on skeleton joint correlation and activity boundary. In *Advances in Visual Computing - 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14-16, 2015, Proceedings, Part II*, pages 489–498, 2015.

[23] Thi-Thanh-Hai Tran, Thi-Lan Le, and J. Morel. An analysis on human fall detection using skeleton from microsoft kinect. In *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*, pages 484–489, July 2014.

[24] Georgios Mastorakis and Dimitrios Makris. Fall detection system using kinects infrared sensor. *Journal of Real-Time Image Processing*, 9(4):635–646, 2014.

[25] Michal Kepski and Bogdan Kwolek. Fall detection using ceiling-mounted 3d depth camera. In *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications, Volume 2, Lisbon, Portugal, 5-8 January, 2014*, pages 640–647, 2014.

[26] Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3):489–501, 2014.

[27] Bogdan Kwolek and Michal Kepski. Improving fall detection by the use of depth sensor and accelerometer. *Neurocomputing*, 168:637–645, 2015.

[28] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2649–2656, 2014.

[29] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 1290–1297, 2012.

[30] Thanh-Hai Tran, Thi-Lan Le, Van-Nam Hoang, and Hai Vu. Continuous detection of human fall using multimodal features from kinect sensors in scalable environment. *Computer Methods and Programs in Biomedicine*, 146:151–165, 2017.

[31] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, Washington, DC, USA, 2011. IEEE Computer Society.

[32] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[33] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297. IEEE, 2012.

[34] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.